

ATTORNEY DOCKET NO.:
069547.0178

PATENT APPLICATION

1

CUSTOMIZABLE TRADING DISPLAY OF MARKET DATA

TECHNICAL FIELD

5 This disclosure relates generally to the field of data processing and, more specifically, to a customizable trading display of market data.

BACKGROUND

Trading systems include one or more workstations that present expanded market data to traders and allow them to offer bids or sales for various financial instruments based on the presented market data. Contemporary trading systems attempt to provide traders with up-to-date market data based on the trader's particular requirements. These trading systems normally are configured prior to or during installation on the trader's workstation. Often, the configurations are static or unchangeable, thereby providing the trader with a fixed display presenting expanded market data.

SUMMARY

This disclosure provides a system and method for providing a customizable trading display of market data. In one embodiment, the method includes selecting a subset from a plurality of quadrants, each quadrant associated with one benchmark instrument and at least one non-benchmark instrument, each non-benchmark instrument associated with the benchmark instrument. Market data is automatically retrieved for the instruments associated with each selected quadrant. The customizable trading display is then generated with the display comprising the subset of quadrants and each quadrant including the associated market data.

In another embodiment, the method further includes receiving a selection of a new benchmark instrument from one of the selected quadrants. A replacement quadrant is selected from the plurality of quadrants, the replacement quadrant associated with the new benchmark instrument. Market data is automatically retrieved for the instruments associated with the replacement quadrant. The customizable display is then updated based on the replacement quadrant, the replacement quadrant including the new market data.

In yet another embodiment, each instrument comprises a market depth and the method further includes generating each quadrant based on a default market depth for each associated instrument. The current market depth of one instrument is updated in response to receiving a market depth update request, the market depth update request comprising a market depth value different from the current market value of the instrument to be updated. The quadrant associated with the updated instrument is then expanded based on the updated market depth. Another quadrant associated with the first quadrant is selected and expanded to match a size of the first quadrant. Then, the remaining quadrants are scaled to maintain a size of the customizable trading display

The invention has several important technical advantages. Various embodiments of the invention may have none, some, or all of these advantages. One advantage of the present invention is that it provides a trading display that may be customized according to a trader's needs or wishes. These customizations may occur at run-time to allow the trader to dynamically filter market data or expand on the

market data currently presented. Further, the present invention may provide a scaleable display that incorporates the trader's customizations without requiring resizing of the entire display or loss of content. Other technical advantages of the present invention will be readily apparent to one skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present disclosure and its advantages, reference is now made to the following descriptions, taken in conjunction with the accompanying drawings, in which:

5 FIGURE 1 illustrates an example trading system for providing a customizable trading display of market data in accordance with one embodiment of the present disclosure;

 FIGURE 2A illustrates an example trading display as might be presented by the trading system of FIGURE 1;

10 FIGURES 2B-C illustrate example trading quadrants with expandable market depth in accordance with the system in FIGURE 1;

 FIGURES 3A-D illustrate an example trading quadrant including selectable instruments in accordance with the system in FIGURE 1;

15 FIGURE 4 is a flowchart illustrating an example method for providing a customizable trading display of market data in accordance with one embodiment of the present disclosure; and

 FIGURES 5A-B are flowcharts illustrating example methods of processing dynamic customizations to the trading display by a trader.

DESCRIPTION OF EXAMPLE EMBODIMENTS

FIGURE 1 is a block diagram illustrating a trading system 100 including one or more customizable trading displays 116 according to one embodiment of the present disclosure. At a high level, trading system 100 is a client/server environment comprising at least one client 102, a server 104, and one or more data providers 106, but may also be a standard computing environment or any other suitable environment. In general, trading system 100 comprises a system for providing a trader, or other user of client 102, with the ability to customize trading display 116 of market data 118, normally received from one data provider 106 and stored in a central repository 110 on server 104. For example, trading system 100 may comprise a market system that compiles U.S. Treasury benchmark and non-benchmark instrument data, automatically presents the compiled instrument data to a trader via customizable display 116, and dynamically processes any appropriate customization of trading display 116 based on requests from the trader. Generally, the benchmark instrument is the most recently auctioned issue for one of the maturities. The term “automatically,” as used herein, generally means that the appropriate processing is substantially performed by at least part of trading system 100. It should be understood that “automatically” further contemplates any suitable user or trader interaction with system 100 without departing from the scope of this disclosure. The term “dynamically,” as used herein, generally means that certain processing is determined, at least in part, at run-time based on one or more variables.

Server 104 comprises any computer including a central repository 110 and communicably connected to at least one client 102 and/or one data provider 106. For example, server 104 may be a general-purpose personal computer (PC), a Macintosh, a workstation, a Unix-based computer, a server computer, or any other suitable device. FIGURE 1 only provides one example of computers that may be used with the disclosure. For example, although FIGURE 1 illustrates one server 104 that may be used with the disclosure, system 100 can be implemented using computers other than servers, as well as a server pool. The present disclosure contemplates computers other than general purpose computers as well as computers without conventional operating systems. As used in this document, the term “computer” is intended to

encompass a personal computer, workstation, network computer, or any other suitable processing device. Computer server 104 may be adapted to execute any operating system including UNIX, Windows, or any other suitable operating system so long as server 104 remains communicably connected to client 102. According to one
5 embodiment, server 104 may be a remote web server. In short, server 104 may comprise any computer with software and/or hardware in any combination suitable to present client 102 with market data stored in a central repository 110. Central repository 110 includes a variety of market data 118 from one or more data providers 116. Central repository 110 comprises any physical or logical description of data
10 storage operable to be defined, processed, or retrieved by externally implemented code. For example, central repository 110 may comprise one or more eXtensible Markup Language (XML) tables or documents. In another example, central repository 110 may comprise a relational database described in terms of SQL statements or scripts, flat files, Btrieve files, or comma-separated-value (CSV) files.
15 Central repository 110 may comprise a plurality of tables stored on one server 104 or across a plurality of servers 104. Moreover, central repository 110 may be local or remote without departing from the scope of this disclosure. In short, central repository 110 is any data storage comprising at least a subset of market data 118 that is communicably connected to at least one client 102.

20 Each client 102 is any trading workstation or device operable to present the trader with customizable trading display 116 via a graphical user interface (GUI). At a high level, illustrated client 102 includes GUI 116, memory 120 and processor 125 and comprises an electronic computing device operable to receive, transmit, process and store any appropriate data associated with trading system 100. It will be
25 understood that there may be any number of clients 102 coupled to server 104, as illustrated by client 102a and client 102b. Further, "client 102" and "user of client 102" may be used interchangeably without departing from the scope of this disclosure. As used in this document, client 102 is intended to encompass a personal computer, workstation, network computer, kiosk, wireless data port, personal data
30 assistant (PDA), one or more processors within these or other devices, or any other suitable processing device. For example, client 102 may comprise a computer that

includes an input device, such as a keypad, touch screen, mouse, or other device that can accept information, and an output device that conveys information associated with the operation of server 104 or clients 102, including digital data, visual information, or customizable display 116. Moreover, client 102 may comprise a keyboard
5 customized to match one particular trading display 116, with each key mapped to a particular frame, view, or functionality of trading display 116. Both the input device and output device may include fixed or removable storage media such as a magnetic computer disk, CD-ROM, or other suitable media to both receive input from and provide output to users of clients 102 through the customizable trading display,
10 namely GUI 116.

GUI 116 comprises a graphical user interface operable to allow the user of client 102 to interface with system 100 to receive one or more sets of market data, each from one data provider 106. Generally, GUI 116 provides the user of client 102 with an efficient and user-friendly presentation of data provided by trading system
15 100. GUI 116 may comprise a plurality of customizable frames or views having interactive fields, pull-down lists, and buttons operated by the user. In one embodiment, GUI 116 presents a trading display that presents the various market data feeds and receives commands from the user of client 102 via one of the input devices. As illustrated in more detail in FIGURE 2A, according to certain embodiments,
20 trading display 116 includes trading grids, trading quads or quadrants 134, a command line, and a ticker. Moreover, it should be understood that the term graphical user interface may be used in the singular or in the plural to describe one or more graphical user interfaces and each of the displays of a particular graphical user interface. Further, trading display 116 contemplates any graphical user interface, such
25 as a generic web browser, that processes information in system 100 and efficiently presents the information to the user. Server 104 can accept data from client 102 via the web browser (e.g., Microsoft Internet Explorer or Netscape Navigator) and return the appropriate HTML or XML responses using network 108.

Network 108 facilitates wireless or wireline communication between computer
30 server 104 and any other computer. Network 108 may communicate, for example, Internet Protocol (IP) packets, Frame Relay frames, Asynchronous Transfer Mode

(ATM) cells, voice, video, data, and other suitable information between network addresses. Network 108 may include one or more local area networks (LANs), radio access networks (RANs), metropolitan area networks (MANs), wide area networks (WANs), all or a portion of the global computer network known as the Internet, and/or any other communication system or systems at one or more locations. Client 102 includes interface 112 for communicating with other computer systems, such as server 104, over network 108 in a client-server or other distributed environment. In certain embodiments, client 102 receives market data 118 from network 108 for storage in memory 120. Generally, interface 112 comprises logic encoded in software and/or hardware in a suitable combination and operable to communicate with network 108 via link 114. More specifically, interface 112 may comprise software supporting one or more communications protocols associated with communications network 108 or hardware operable to communicate physical signals.

Memory 120 may include any memory or database module and may take the form of volatile or non-volatile memory including, without limitation, magnetic media, optical media, random access memory (RAM), read-only memory (ROM), removable media, or any other suitable local or remote memory component. In the illustrated embodiment, memory 120 includes at least trade application 130, local market data 132, and quadrants 134, but may also include any other appropriate data.

Trade application 130 could include any hardware, software, firmware, or combination thereof operable to receive and process market data 118 and present it to the trader through customizable trading display 116. Further, trade application 130 may receive and process customization requests from the trader using client 102 via trading display 116. It will be understood that while trade application 130 is illustrated as a single multi-tasked module, the features and functionality performed by these engine may be performed by multiple modules such as, for example, a data retrieval module and a presentation engine. Moreover, trade application 130 may comprise a child or sub-module of another software module without departing from the scope of this disclosure. In short, trade application 130 comprises one or more software modules operable to provide any appropriate backend processing for customizable trading display 116.

Local market data 132 stores one or more subsets of remote market data 118, such as benchmark and non-benchmark instruments. Local market data 132 may receive a copy of market data 118 through interface 112 or from another process running on client 102 as appropriate. As appropriate, local market data 132 may automatically, dynamically, or manually retrieve or receive market data 188 from central repository 110 at any appropriate time. For example, trading application 130 may automatically retrieve a copy of market data 118 in central repository 110 upon initial execution. Further, local market data 132 may be of any suitable format including XML documents, flat files, comma-separated-value (CSV) files, SQL tables, relational database tables, and any other format operable to store at least one subset of market data 118. It will be understood that local market data 132 may be in a format different from central repository 110 or communicated market data 118 so long as it is compatible with trading display 116 and may be processed by trading application 130.

Trading display 116 includes a number of quadrants 134 selected from a plurality of quadrants 134 stored in memory 120. Each quadrant 134 is operable to display expanded trading information associated with one benchmark instrument. Each quadrant 134 may also include one or more non-benchmark instruments that may be selected, for example, in response to the trader pressing one of the issue keys on the keyboard. One quadrant 134 may also provide the trader with the ability to obtain trading information for each of the related U.S. Treasury swaps, off-the-runs, yield curve U.S. Treasury swaps, basis, or any other appropriate benchmark instrument.

Client 102 also includes processor 125. Processor 125 executes instructions and manipulates data to perform the operations of client 102 such as, for example, a central processing unit (CPU), an application specific integrated circuit (ASIC) or a field-programmable gate array (FPGA). Although FIGURE 1 illustrates a single processor 125 in client 102, multiple processors 125 may be used according to particular needs, and reference to processor 125 is meant to include multiple processors 125 where applicable. In certain embodiments, processor 125 executes

one or more processes associated with trade application 130 to present market data 118 communicated from central repository 110 via data providers 106.

Data provider 106 typically comprises a third party web server or an enterprise agent residing on a machine operable to communicate at least a portion of market data 118 for storage in central repository 110. It will be understood that data provider 106 may be remote or on-site; further, data provider 106 may represent a separate process running on server 104 or client 102 without departing from the scope of this disclosure. Generally, data provider 106 is any hardware, software, or logic operable to provide trading system 100 with at least a subset of any appropriate market data 118. Trading system 100 may comprise any number of data providers 106, as illustrated by data providers 106a and 106b, as long as central repository 110 includes at least a portion of desired market data 118. For example, a plurality of data providers 106 may be communicably daisy-chained off a master provider 106. Master data provider 106 may compile and verify substantially all of market data 118 and communicate the compiled market data 118 to server 104 upon request or at a scheduled time.

In one aspect of operation, trading application 130 is executed and initialized either automatically, such as when client 102 is powered up, or in response to a command from the trader. Trading application 130 determines a default configuration and layout for trading display 116, retrieves the appropriate market data for the benchmark and non-benchmark instruments from central repository 110 and/or local market data 132, and generates customizable trading display 116 based on the default configuration and retrieved data. The generated trading display 116 includes a plurality of customizable quadrants 134. For example, trading display 116 allows the trader to expand the viewable market depth in at least one of quadrants 134, select and replace various instruments in quadrants 134, or any customize according to any other appropriate criteria. Trading application 130 processes each customization and regenerates, or repaints, trading display 116 as appropriate.

FIGURE 2A illustrates example customizable trading display 116. Illustrated trading display 116 includes a plurality of trading grids, a plurality of trading quads or quadrants 134, a command line, and a ticker. Each trading grid presents market data

and market updates on selected securities. From the trading grid, the trader can post markets and execute trades. Trading display 116 may display grids in a tabbed format or multiple panes of adjustable width. If necessary, horizontal scroll bars appear to allow the user to navigate the grid. Further, each row can be expanded to display the
5 bid/offer list by selecting the arrow on the left of each row in a grid. Trading display 116 includes one or more customizable quadrants 134 based on one of the cells in the grid. Each quadrant 134 is typically associated with a default layout determined at initialization of trading application 130. This default layout may be dynamically configured by the trader during processing. Illustrated display 116 includes four
10 quadrants 134. But trading display 116 may include any number of quadrants 134 so long as display 116 remains appropriate. Using quadrants 134, the trader may view the benchmark and non-benchmark instruments associated with the selected market cell. As described above, each quadrant 134 may be linked to one set of keys on the customized keyboard designating bid, offer, buy, sell, cancel bid/offer, and others.
15 Trading display 116 may also include the command line that may be used to enter orders using the number keys and keyboard shortcuts.

Status tabs may include active orders, trade history, market history, overall application status, or any other suitable status information. The active orders window displays information on all posted markets or unexecuted orders. This window
20 displays information on all currently outstanding bids and offers as well as information on buys and sells that have not been confirmed. Double clicking on an entry in this window will display a dialog that offers the option of canceling the selected entry. The trade history window displays information on all confirmed trades. The query trades command may be used by the trader to refresh the contents
25 of this window and trading history settings to specify the dates that the query trades command should specify. The market history window displays information on all markets and orders rejected by the trading system. The overall application status window contains a log of communications between components of trading system 100 as well as a record of actions related to trading functions. According to certain
30 embodiments, the contents may be communicated to a file or printer by right clicking in view 134. The ticker feature typically displays the time and selected securities on a

banner that scrolls across the screen. These example graphical features are for illustration purposes only and trading display 116 comprises quadrants as well as none, some, or all of the remaining illustrative elements or alternative elements without departing from the scope of the disclosure. It will be understood that
5 illustrated display 116 is for example purposes only and trading display 116 may comprise any appropriate graphical elements for presenting appropriate market data 118. Moreover, it will be understood that the data displayed therein is for example purposes only and any appropriate market data 118 may be displayed.

FIGURES 2B and 2C illustrate example quadrant 134 including at least one
10 instrument with expandable market depth. Illustrated quadrant 134 is a two year U.S. Treasury expanded view including benchmark instrument 205 and a plurality of non-benchmark instruments 215. According one embodiment, quadrant 134 further includes a headline. Headline may include the price that should be initially aggressed to start a trade or the actual aggressed trade. The headline may further include the
15 item, the price, the size, and the last price for the benchmark instrument. Quadrant 134 may include bid and offer stacks without departing from the scope of this disclosure. Bid and offer stacks may indicate the size of one or many market participants that are bidding or offering at the current price.

Under the benchmark instrument 205, trading application 130 may provide the
20 trader with U.S. Treasury swap lock price (*e.g.*, “L 99.26”) on the benchmark instrument. U.S. Treasury swap trading relates to a chosen price movement relationship between two different instruments. Quadrant 134 may also include any other suitable trading information, such as market depth information, which shows price and size for items that have not yet been traded. Illustrated U.S. Treasury swap
25 lock price is L 99.26. It will be understood that, for example, U.S. Treasury bonds are traded at a percentage of a bond’s face value. This percentage is typically broken down into 32nds of a point and fractions thereof. Often, the displayed prices shown represent the whole number portion and the percentage being traded. For example, consider the price “99.256.” The number left of the decimal point, “99”, is equivalent
30 to the whole number portion of the price and the number right of the decimal point, “256”, is equivalent to the fractional portion. According to certain embodiments,

some prices may have “+” in the third digit right of the decimal point. The “+” may represent 4/8th of a 32nd or half a 32nd.

Each instrument includes a market depth comprising a value between one and ten. Each instrument in quadrant 134 may further have an upper and lower limit on the market depth that may be displayed. For example, the trader may set the required upper and lower ranges within the range of one and ten. Moreover, each displayed benchmark and non-benchmark instrument is associated with one identifiable GUI element 210 such as an arrow, a plus (+) sign, or any other GUI component operable to notify the user that the instrument may or may not have market depth. For example, illustrated quadrangle 134 includes a shaded arrow 210 associated with the benchmark instrument. According to certain embodiments, shaded arrow 210 (pointing to the right) signifies that the instrument has market depth greater than one and, therefore, can be expanded. Each unshaded arrow 210 (pointing to the right) signifies that the associated instrument has no market depth, or market depth equal to one, but can be expanded. For example, if the trader clicks on arrow 210 (or presses a key or a key combination on the customized keyboard), the market depth of the instrument is displayed. The trader will be able to define the amount of depth shown when the instrument is expanded. Once expanded to the requested value, arrow 210 will typically point down, as illustrated in FIGURE 2C. Typically, expanded quadrant 134 will remember the market depth layout for subsequent generation. If the market depth of one of the instruments is changed, often trading application 130 will modify the instrument at the same position in other quadrants 134 on the same row in display 116 to keep the view symmetrical. It will be understood that arrows 210 are for illustration purposes only and that any GUI element may be used without departing from the scope of the disclosure.

FIGURES 3A-D illustrate an example trading display 116 including selectable instruments in accordance with trading system 100. The illustrated data and layout thereof are similar to that in FIGURES 2B-C. As illustrated in FIGURES 3A-D, quadrant 134 provides the trader with the ability to select various instruments for display with quadrant 134. For example, the trader may select a different benchmark instrument, after which trading application 130 retrieves data for the replace

benchmark instrument and all default associated non-benchmark instruments and regenerates the entire quadrant 130 based on this retrieved data. In another example, the trader may select a replacement non-benchmark instrument. In this example, trading application 130 may retrieve the data associated this instrument alone from
5 local market data 132 or central repository 110 as appropriate. Next, trading application 130 repaints quadrant 134.

According to certain embodiments, to select a replacement instrument, the trader presses the right mouse button over the selected instrument to access a popup window, as illustrated in FIGURE 3B. The trader then selects "Change Instrument"
10 from the popup window, which then provides a list of replacement instruments. It will be understood that each changeable instrument may have a list of none, one, or more possible replacements. The trader selects one replacement instrument from the list provided. Typically, no duplicate instruments are allowed on the same quadrant 134 or display 116. Then trading application 130 regenerates or repaints the effected
15 quadrant 134 as appropriate. But the default view of quadrant 134 may be reverted to at anytime without departing from the scope of this disclosure

It will be understood that FIGURES 2B-C and 3A-D are for illustration purposes only. Each illustrates one or more example outcomes of a particular customisation technique such as, for example, expansion of market depth and
20 selection of replacement instruments. While illustrated separately, these techniques may be implemented alternatively or in combination within one or more trading displays 116, one or more quadrants 134, or any other appropriate functional element.

FIGURE 4 is a flowchart illustrating an example method 400 for providing a customizable trading display 116 of market data in accordance with one embodiment
25 of the present disclosure. Generally, method 400 describes trading client 102 initializing, generating, and presenting trading display 116 to a trader. Method 400 is described in respect to system 100 and trading application 130. However, any other suitable system may use appropriate embodiments of method 400 to present customizable trading display 116 to the trader without departing from the scope of
30 this disclosure.

Method 400 begins at step 402, where trading application 130 selects a subset of quadrants 134 from memory 120. It will be understood that this selection may occur automatically through trading application 130 determining a default configuration, receiving a selection from the trader using client 102, or any other appropriate selection technique. Next, at step 404, trading application 130 associates one benchmark instrument with each selected quadrant 134. Trading application 130 then queries central repository 110 based on each determined benchmark at step 406. The query from trading application 130 may include a request for market data 118 associated with each benchmark and market data 118 for each non-benchmark instrument associated with the selected benchmark. Trading application 130 may query local market data 132 alternatively or in combination with central repository 110 at step 406. At step 408, trading application 130 receives benchmark data for each selected view from central repository 110. Trading application 130 receives non-benchmark instrument data associated with each benchmark at step 410. Once the views have been selected and the appropriate data has been compiled, trading application then generates the customizable trading display 116 in steps 414 through 434.

Trading application 130 begins generating trading display 116 by selecting a first expanded view or quadrant 134 from the selected subset at step 412. Next, at step 414, trading application 130 initializes the market depth of each instrument in the selected quadrant 134. According to certain embodiments, trading application 130 may determine the market depth based on a stored configuration, a default value, or any other appropriate parameter. Next, at decisional step 416, trading application 130 determines if the benchmark associated with the selected view 134 has an available market depth greater than one. If the benchmark has an available market depth greater than one, then trading application 130 generates an identifiable GUI element 210, such as a shaded or colored arrow or plus sign, to represent expandable market depth associated with the benchmark at step 418. Next, or if the benchmark does not have an available market depth greater than one, processing proceeds to step 420, where trading application 130 selects the first non-benchmark instrument associated with the benchmark. At decisional step 422, trading application 130 determines if the

non-benchmark instrument associated with the selected view 134 has an available market depth greater than one. If the non-benchmark instrument has an available market depth greater than one, then trading application 130 generates an identifiable GUI element 210 to represent expandable market depth associated with the non-benchmark instrument at step 424.

Trading application 130 then determines if there are more non-benchmark instruments associated with the benchmark at decisional step 426. If there are more non-benchmark instruments, then trading application 130 selects the next non-benchmark instrument in the selected quadrant 134 at step 427 and processing returns to decisional step 422. Once there are no more non-benchmark instruments in the selected view 134, then processing proceeds to decisional step 428. At decisional step 428, trading application 130 determines if there are more quadrants 134 in the subset previously selected at step 402. If there are more quadrants 134, then trading application 130 selects the next view 134 in the subset and execution returns to step 414. Once all the selected quadrants 134 have been initially processed, then trading application 130 generates or paints each selected quadrant 134 at step 430 based on market data received in steps 408 and 410 and the settings determined in steps 414 through 424. Next, at step 432, trading application 130 generates customizable display 116 based on the generated views. Once the customizable trading display 116 has been generated, trading application 130 presents trading display 116 to a trader or other appropriate user of client 102.

FIGURES 5A-B are flowcharts illustrating example methods 500 and 550 for processing dynamic customizations to trading display 116 by a trader. Generally, methods 500 and 550 describes trading client 102 receiving customization requests from the trader, processing any suitable customizations, and presenting an updated trading display 116 based upon the processed customizations. Methods 500 and 550 are described in respect to trading system 100. However, any other suitable system may use different embodiments of methods 500 and 550 to customize trading display 116 without departing from the scope of this disclosure.

FIGURE 5A illustrates an example method 500 for changing one view 134 and trading display 116 according to certain embodiments of the disclosure. Method

500 begins at step 502, where trading application 130 receives a request to change a benchmark from a trader at a first quadrant, or expanded view, 134 and display 116. At decisional step 504, trading application 130 determines if the requested benchmark is already represented in trading display 116. If the benchmark is already represented, then trading application 130 communicates an error message to the trader via display 116 at step 506. Otherwise, trading application 130 selects a replacement view or quadrant 134 based on the requested benchmark at step 508. Next, at step 510, trading application 130 queries central repository 110 based on the requested benchmark. According to certain embodiments, trading application 130 may query local market data 132 alternatively or in combination with central repository 110. At step 512, trading application receives the market data for the benchmark associated with the replacement quadrant 134. Trading application 130 then receives market data for one or more non-benchmark instruments associated with the replacement benchmark at step 514. Next, at step 516, trading application 130 generates the replacement view 134 based on the received data. During the generation process, trading application 130 may initialize the current market depth for each instrument in the replacement view 134. Moreover, trading application 130 may generate one identifiable GUI element for each instrument that includes an available market depth greater than one. Trading application 130 then replaces the first view with the generated replacement view at step 518. At step 520, trading application 130 regenerates, or repaints, customizable trading display 116 based on the replacement view and the views remaining in display 116. Once display 116 has been regenerated, then trading application presents the updated customizable trading display 116 to the trader or other appropriate user at step 522.

FIGURE 5B illustrates an example method 550 for updating the viewable market depth of one of the instruments in trading display 116. Example method 550 begins at step 552, where trading application 130 receives a selection of one of the instruments in a first view in trading display 116. Next, at step 554, trading application 130 receives a request to display the available market depth at the selected instrument via an associated identifiable GUI element 210. As illustrated in FIGURES 2B-C, this identifiable GUI element 210 may comprise an arrow of a

particular color or shading. Trading application 130 then determines if the selected instrument has any available market depth at decisional step 556. If not, then processing ends, trading application 130 communicates an error message to the trader, or any other appropriate response to an invalid request occurs. Otherwise, trading application 130 expands the first quadrant 134 to include the requested market depth for the selected instrument at step 558. Trading application 130 then determines if there is a second quadrant 134 associated with first view 134 at decisional step 560. For example, first and second quadrants 134 may be substantially adjacent in display 116. If two or more quadrants 134 are associated, then trading application 130 expands the size of at least the second view 134 to match expanded first quadrant 134 at step 562. Part of this expansion process may include trading application 130 scaling the remaining views 134 and display 116 to maintain an appropriate size of display 116. Next, at step 564, trading application 130 stores the current market depth sentence for each instrument and the views 134 for subsequent generation. Once the appropriate views 134 have been expanded and, as appropriate, scaled, then trading application 130 regenerates display 116 based on the updated views 134 and presents display 116 to the trader.

The preceding flowcharts focus on the operation of example trading system 100 described in FIGURE 1, as this diagram illustrates functional elements that provide for the preceding techniques for presenting and processing customizable trading display 116. However, as noted, trading system 100 contemplates using any suitable combination and arrangement of functional elements for providing these operations, and these techniques can be combined with other techniques as appropriate. Further, various changes may be made to the preceding flowcharts without departing from the scope of this disclosure. For example, many of the steps in these flowcharts may take place simultaneously and/or in different orders than as shown. Moreover, trading system 100 may implement methods with additional steps, fewer steps, and/or different steps, so long as the methods remain appropriate.

Although this disclosure has been described in terms of certain embodiments and generally associated methods, alterations and permutations of these embodiments and methods will be apparent to those skilled in the art. Accordingly, the above

description of example embodiments does not define or constrain this disclosure. Other changes, substitutions, and alterations are also possible without departing from the spirit and scope of this disclosure.